



ELSEVIER

Comput. Methods Appl. Mech. Engrg. 154 (1998) 203–228

**Computer methods
in applied
mechanics and
engineering**

Physics based GMRES preconditioner for compressible and incompressible Navier–Stokes equations

N. Nigro^{a,*}, M. Storti^{a,1}, S. Idelsohn^{a,2}, T. Tezduyar^{b,3}

^a*Grupo de Tecnología Mecánica del INTEC CONICET—Universidad Nacional del Litoral, Güemes 3450, 3000 Santa Fe, Argentina*

^b*Aerospace Engineering and Mechanics, Army HPC Research Center, University of Minnesota, 1100 Washington Avenue South, Minneapolis, MN 55415, USA*

Received 11 January 1996

Abstract

This paper presents the implementation of a local physics preconditioning mass matrix [8] for an unified approach of 3D compressible and incompressible Navier–Stokes equations using an SUPG finite element formulation and GMRES implicit solver. During the last years a lot of effort has been dedicated to finding a *unified approach* for compressible and incompressible flow in order to treat fluid dynamic problems with a very wide range of Mach and Reynolds numbers [10,26,37]. On the other hand, SUPG finite element formulation and GMRES implicit solver is one of the most robust combinations to solve state of the art CFD problems [1,6,9,22,29,30,31].

The selection of a good *preconditioner* and its performance on parallel architecture is another open problem in CFD community. The *local* feature of the preconditioner presented here means that no communication among processors is needed when working on parallel architectures. Due to these facts we consider that this research can make some contributions towards the development of a unified fluid dynamic model with high rates of convergence for any combination of Mach and Reynolds numbers, being very suitable for massively parallel computations.

Finally, it is important to remark that while this kind of preconditioning produces stabilized results in nearly incompressible regimes the standard version exhibits some numerical drawbacks that lead to solutions without physical meaning. © 1998 Elsevier Science S.A.

1. Introduction

A very common procedure in order to find steady states from nonlinear equations arising in computational fluid dynamics (CFD), say $G(x) = 0$, is to iterate an explicit or implicit temporal scheme until convergence: $M\dot{x} = G(x)$, $t \rightarrow \infty$. Here, $x \in \mathbb{R}^N$ is the state vector, G is map from \mathbb{R}^N onto itself that represents the discrete system of equations, M is the mass matrix of the system and the dot represents time derivative. This kind of procedure is called *time marching* and is widely used for the computation of compressible subsonic, transonic, supersonic and hypersonic regimes, either in inviscid or viscous flows [1–4,15–17,22,23,28].

The inconvenience appears when we need to solve low subsonic or incompressible flow and in this case we need to switch to an incompressible formulation [5,6,29,32,33] or to use some perturbation method [14,18,19] to get a solution. However, there are a lot of interesting physical applications that contain several different regimes inside the domain like compressible inviscid flow in the core, compressible viscous flow around solid objects and nearly incompressible flow in some other parts. Some challenge problems of this type can be founded in flow through strongly converging–diverging nozzles used in propulsion, stagnation points and large recircula-

* Corresponding author.

¹ Research Staff member of CONICET Argentina.

² Professor at Universidad Nacional del Litoral.

³ Professor at University of Minnesota.

tion zones, flow around turbomachinery and helicopter rotors where the rotational velocity goes from nearly incompressible flow close to the axis (root) to high subsonic, transonic or supersonic flow in the outer part of the rotor.

To circumvent these difficulties, unified formulations have been proposed that allow to treat efficiently these extreme situations [10,25,26,37].

Following with time marching schemes, these are driven by explicit or implicit methods where core memory and CPU-time are two of the most important factors to take into account for its selection. If an explicit scheme is used, comparatively low amount of core memory and large CPU-time are required. Moreover, the CPU-time highly depends on the conditioning of the system. Bad conditioning of the system is caused by several multiplicative factors like: large variations in element size through the mesh, large variations in edge sizes for a given element, local incompressible ($M \rightarrow 0$) or transonic ($M \rightarrow 1$) regimes, where M is the Mach number, strong viscous zones with recirculation and so on. On the other hand, if an implicit temporal scheme coupled to a direct solver is preferred, high convergence rates are achieved, but it requires a large amount of memory to factorize the associated matrix. This problem can be partially overcome by solving the linear subproblems by an iterative 'black-box' solver like GMRES [20,21]. Specially with the matrix-free version [1,22] the amount of core memory is drastically reduced at the expense of an increased CPU-time which, again, highly depends on the conditioning of the system. So, the success of an iterative scheme (either explicit or implicit) is related to improving the conditioning of the system. For instance, bad conditioning arising from large variations in mesh element size can be removed through the use of 'local time stepping' strategies [34]. This can be seen as modifying the mass matrix of the original system to $\tilde{M}\dot{x} = G(x)$ where \tilde{M} is a diagonal matrix that includes a factor proportional to the maximum admissible time step based on a local stability analysis. This strategy is named *preconditioning mass matrix* (PMM). As is well known, this modifies the temporal evolution of the unknowns in such a way that it has no more physical sense. Due to this fact, it is only applicable to steady state simulations. With the same philosophy in mind we are looking for a *local physics based preconditioning mass matrix* which correct the bad conditioning associated to different characteristic speeds at some specific regimes. By 'local' mass matrices we mean block diagonal matrices with each block connecting only the d.o.f.'s of each node. This restriction is imposed in order to have an $O(N_{\text{nod}})$ cost per preconditioning cycle, where N_{nod} is the total number of nodes in the mesh.

The goal of PMM is to improve the condition of the system by a characteristic speeds redefinition of different waves present in the physical problem keeping some necessary properties unaltered, for example the hyperbolicity in inviscid flow. The selection of the preconditioner is not an easy task. There are several ways to find it, from mathematical arguments based on discrete Fourier transform, from a physical point of view by group velocity analysis or by heuristic arguments. Using physical arguments we define the *group velocity condition number* (κ) = $|v_G|_{\text{max}} / |v_G|_{\text{min}}$, where $|v_G|_{\text{max,min}}$ are the maximum and minimum group velocities of the discrete scheme [27,36].

This number represents the ratio between the maximum and minimum speed of each wave arising in the physical problem. In 3D compressible flow we have five different modes, *entropy waves*, *acoustic waves* and *vorticity waves*, each one traveling with its own speed through the domain. If we consider those waves as error waves we need to accelerate them to the boundaries and to absorb them on it or to dissipate inside the computational domain. These two mechanisms represent mainly the two ways to annihilate error components during the computation. For inviscid flow, where the diffusion exists only numerically, we need to equalize as much as possible the different wave speeds as an optimal criterion. In this sense the preconditioner redefines the acoustic speed close to the particle speed circumventing the stiffness associated with nearly incompressible ($\kappa = 1/M$) or transonic flow ($\kappa = 2/|M - 1|$) keeping the good behavior of subsonic and supersonic regimes. This problem was solved using different preconditioner candidates and there are several papers reporting good results [14,18,19,25,27].

The extension to viscous flow is not so straightforward because of the introduction of dissipation effects. This changes the mathematical character of the system going from hyperbolic to parabolic in some regions where the viscous term dominates. Viscous and acoustic time scales become dominant, and the good conditioning depends on the balance of these two scales. It was natural to solve this problem changing the formulation for another more appropriate like those based on pressure instead of density. This is a very important difference between preconditioners for inviscid problems where the *density based method* is much used and preconditioners for viscous flow where the *pressure based method* is preferable.

In this sense we adopt the preconditioner presented by Choi et al. [8,35] that was deduced from perturbation methods and implemented as a PMM method and it was proved to be efficient to a very wide and representative range of Mach and Reynolds numbers. In the original paper the authors have used a finite volume formulation and an ADI scheme to solve the equations system in 2D [8]. Here, we extend to 3D simulations with an SUPG finite element formulation using a matrix-free version of GMRES implicit solver that represents one of the most used combinations to solve CFD problems nowadays.

During recent years, a lot of effort was done to find good preconditioners to GMRES method, most of them designed with the goal of finding good approximations for the inverse of the system matrix. Preconditioners like SSOR, ILU with different level of filling, have proved to be relatively efficient and partially applicable. Moreover, they have intrinsic difficulties to be parallelized.

The *local* PMM presented here has shown to be efficient for a wide range of Mach and Reynolds numbers and also suitable for parallel codes because they do not need communication among processors. Also, it gives some answer to the problem of searching a good GMRES preconditioner for the solution of compressible and incompressible Navier–Stokes equations. Finally, it is important to remark that using a standard SUPG formulation in nearly incompressible flow the numerical solution exhibits some inconveniences associated with the stabilization of the scheme leading to non-physical solutions. For example, using a very low Mach number ($M < 10^{-2}$) spurious oscillations have been detected in regions with sharp variations in the solution. In some other problems we have found diverging behavior in time or convergence to a numerical solution with no physical sense (very large density perturbations). So, this preconditioner is essential to get accurate solutions in nearly incompressible flows and is very effective to accelerate the convergence for all Mach and Reynolds numbers.

In the next section we present the physical and mathematical model of our problem introducing the preconditioner on the temporal term. In Section 3 we include the variational formulation of our SUPG finite element method. Section 4 introduces the preconditioner used in this research and Section 5 shows some numerical topics related with the implementation. In Section 6 we add several numerical examples to test the behavior of this preconditioner applied to a matrix-free GMRES implicit solver. Finally, Section 7 presents the conclusions and future trends of this formulation.

This research is oriented to the further application of the results presented here to simulate flow around helicopter and turbomachinery rotors.

Register for free at <https://www.scipedia.com> to download the version without the watermark

2. Definition of the problem

The compressible Navier–Stokes equations modified by the introduction of a preconditioner on the time derivative term is written as

$$\Gamma_v \frac{\partial Q_v}{\partial t} + A \cdot \nabla Q = K \nabla^2 Q + F \quad (1.1)$$

The preconditioner Γ_v introduced in the temporal term will be presented later.

There are two different variables involved in this problem, the viscous variables Q_v and the conservative variables Q , defined as

$$Q = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e \end{pmatrix} \quad Q_v = \begin{pmatrix} p \\ u \\ v \\ w \\ T \end{pmatrix} \quad (1.2)$$

The viscous variables are necessary to get a good condition number in the viscous limit. Some details about this subject are presented in Section 4.

As is well known, this system introduces the inertial forces through the advective Jacobians (A), the viscous forces through the diffusive tensor (K) and source terms (F).

If we choose the preconditioner $\Gamma_v = \partial Q / \partial Q_v$ the above system becomes the standard *Navier–Stokes equations* and for a general Γ_v we have to restrict our selection in order to reach the same steady solution as the

standard formulation. Then, this kind of preconditioner is adequate only for steady state problems. Extensions to treat unsteady phenomena are feasible considering each time step like a steady problem where the solution at t_n is the initial condition and the solution at t_{n+1} plays the role of steady solution in the interval $[t_n, t_{n+1}]$.

This physical model is closed by the energy equation and the ideal gas law

$$\rho e = \frac{p}{\gamma - 1} + \frac{1}{2} \rho \|u\|^2, \quad p = \rho RT \quad (1.3-4)$$

where $R = (\gamma - 1)C_v$ is a universal gas constant and C_v is the specific heat at constant volume.

Finally, the mathematical model is well posed by the introduction of adequate initial and boundary conditions.

3. Variational formulation

Now, we present the variational formulation of the above system of equations associated with the SUPG method.

First, we transform the above system of equations in another one with only one kind of unknown, the conservative variables. Then

$$\Gamma \frac{\partial Q}{\partial t} + A \cdot \nabla Q = K \nabla^2 Q + F \quad (2)$$

where $\Gamma = \Gamma_v \partial Q_v / \partial Q$ is the preconditioner is the conservative basis.

The finite element formulation of SUPG applied to the preconditioned problem is written as follows:

Find $Q^h \in \mathcal{S}^h$ such that $\forall W^h \in \mathcal{V}^h$

$$\begin{aligned} & \int_{\Omega} W^h \cdot \left(\Gamma \frac{\partial Q^h}{\partial t} + A_i^h \frac{\partial Q^h}{\partial x_i} \right) d\Omega + \int_{\Omega} \frac{\partial W^h}{\partial x_i} \cdot K_{ij} \frac{\partial Q^h}{\partial x_j} d\Omega \\ & + \sum_{e=1}^{\text{numel}} \int_{\Omega^e} (\tau')^T (A_k^h)^T \frac{\partial W^h}{\partial x_k} \cdot \left\{ \Gamma \frac{\partial Q^h}{\partial t} + A_i^h \frac{\partial Q^h}{\partial x_i} - \frac{\partial}{\partial x_i} \left(K_{ij} \frac{\partial Q^h}{\partial x_j} \right) - F \right\} d\Omega \\ & + \sum_{e=1}^{\text{numel}} \int_{\Omega^e} \delta'_{shc} \frac{\partial W^h}{\partial x_i} \frac{\partial Q^h}{\partial x_i} d\Omega = \int_{\Omega} W^h F d\Omega + \int_{\partial\Omega_g} W^h f d\partial\Omega \end{aligned} \quad (3)$$

where

$$\begin{aligned} \mathcal{S}^h &= \{Q^h \mid Q^h \in [H^{1h}(\Omega)]^{\text{ndf}}, Q^h|_{\Omega^e} \in [P^1(\Omega^e)]^{\text{ndf}}, Q^h = g \text{ on } \partial\Omega_g\} \\ \mathcal{V}^h &= \{W^h \mid W^h \in [H^{1h}(\Omega)]^{\text{ndf}}, W^h|_{\Omega^e} \in [P^1(\Omega^e)]^{\text{ndf}}, W^h = 0 \text{ on } \partial\Omega_g\} \end{aligned}$$

REMARK. This formulation, without Γ , and with matching definitions for τ' and δ'_{shc} , was given in [2,3,15,16], which, in turn, are all based on the formulation introduced in [28] plus a shock-capturing term.

The first two integrals on the left-hand side represent the Galerkin contribution and the following two come from the SUPG perturbation function added to the standard weight function, introducing the necessary numerical diffusion to stabilize the scheme. δ'_{shc} is used as shock capturing operator necessary when very sharp discontinuities are present. Note that we want to emphasize the difference between the non-preconditioned formulation with respect to the above preconditioned one including not only the preconditioner matrix Γ but also both parameters τ' and δ'_{shc} where the primes means that the preconditioner affects on their definitions. f and g represent the natural and Dirichlet boundary conditions vectors, respectively.

For more details about SUPG formulation see [2,3,7,11–13,15,16,23,28,32,33].

4. An optimal local preconditioner

Now, we present the preconditioner introduced in Eq. (1.1). This matrix can be found by subtracting the continuity equation from the energy equation using the ideal gas law, transforming to viscous variables (Eq.

(1.2)), changing the mass matrix in order to improve the condition number of the system and finally transforming back to the conservative basis. For details about its computation see [8].

This preconditioner can be written as

$$\Gamma_v = \begin{pmatrix} \frac{1}{\beta M^2} & 0 & 0 & 0 & 0 \\ \frac{u}{\beta M^2} & \rho & 0 & 0 & 0 \\ \frac{v}{\beta M^2} & 0 & \rho & 0 & 0 \\ \frac{w}{\beta M^2} & 0 & 0 & \rho & 0 \\ \frac{(e+p)}{\rho \beta M^2} - \delta & \rho u & \rho v & \rho w & \frac{\gamma p R}{\gamma - 1} \end{pmatrix} \quad (4)$$

This kind of preconditioning matrix has shown to have uniform convergence at all Reynolds and Mach numbers through several numerical experiments.

The introduction of viscous variables instead of conservative ones makes this scheme similar to *pressure based methods* and the role of the preconditioner is to keep the viscous and acoustic scales of the same order when the viscous terms are dominant. On the other hand, when advection dominates it is better to switch to a *density based method* and the goal of the preconditioner is to equalize the particle and the acoustic wave speeds. In order to unify the formulation getting a low condition number for all Mach and Reynolds numbers, we need a preconditioning capable of switching between these two methods according to the local flow conditions.

This switch is carried out by β parameter (Eq. (4)) and following the original definition of Choi et al. [8] we have

$$\beta M_r^2 = \epsilon c^2 \quad \text{with } \epsilon = \max(\epsilon_{\text{inv}}, \epsilon_{\text{vis}}) \quad (5.1-5)$$

$$\begin{cases} \epsilon_{\text{inv}} = M_r^2 \\ \epsilon_{\text{vis}} = \min_j \left[\frac{\alpha_j(\alpha_j - 1)}{(\alpha_j - 1 + c^*/u_j)} \right] \end{cases}$$

$$\alpha_j = \frac{\text{CFL}}{\sigma \text{Re}_{\Delta x_j}} \quad j = x, y, z$$

We temporarily use u_x, u_y, u_z instead of u, v, w for simplicity. M_r is a reference Mach number used to avoid singularities when velocity is locally zero and $M_{\min} = 10^{-6}$ in this work. This value is defined as

$$M_r = \begin{cases} M_{\min} & M < M_{\min} \\ M & M_{\min} < M < 1 \\ 1 & M > 1 \end{cases} \quad (5.6)$$

σ represents the algorithmic Fourier number and $\text{Re}_{\Delta x_j}$ is the element Reynolds number with Δx_j as the characteristic length.

δ is an arbitrary constant that plays the role of a coefficient of the time derivative of pressure. When $\delta = 0$ the time derivative is dropped out and when $\delta = 1$ we recover the standard energy equation.

5. Numerical implementation

Even though the implementation of a modified mass matrix as a preconditioner seems to be straightforward, there are several details to take into account not to deteriorate the convergence rate. In the first part of this section we deal with the definition of the stabilization parameters introduced by SUPG modified by the

preconditioner. Next, we apply the semidiscretized numerical scheme to explicit and implicit solvers and finally we present the implementation of absorbent boundary conditions over the preconditioned system.

5.1. The matrix of intrinsic time scale

As can be seen from (3), the terms introduced by SUPG method are characterized by two different parameters, τ' and δ'_{shc} . While the former globally stabilizes the scheme because of the presence of advection dominated regions, the latter acts locally avoiding the appearance of undershoots in the vicinity of sharp discontinuities. The definitions of both parameters have been one of the most important problems in CFD community during the last fifteen years and there are a lot of important contributions for this topic.

As we have mentioned in the previous section, the presence of the preconditioner affects the computation of these parameters. In this paper we focus on the definition and usage of the global stabilization parameter. The shock capturing parameter (δ'_{shc}) is under development and will be presented in future works.

Starting with (1.1) and multiplying both sides of the equation by the inverse of the preconditioner in conservative basis we obtain the following equation:

$$\frac{\partial Q}{\partial t} + \tilde{A} \cdot \nabla Q = \tilde{K} \nabla^2 Q + \tilde{F} \quad \text{with} \quad \begin{cases} \Gamma = \Gamma_v \frac{\partial Q_v}{\partial Q} \\ \tilde{A} = \Gamma^{-1} A \\ \tilde{K} = \Gamma^{-1} K \\ \tilde{F} = \Gamma^{-1} F \end{cases} \quad (6.1-5)$$

Expressions for some of these matrices are included in Appendix A.

As a standard procedure, the numerical diffusivity introduced by SUPG method in the inviscid case

$$\tilde{K}^{num} = \tilde{A} \tilde{\tau} \tilde{A} \quad (7.1)$$

requires the computation of the intrinsic time matrix $\tilde{\tau}$. In this work we use the definition presented by Hughes et al. [12] adapted to the preconditioned system

$$\tilde{\tau} = \|\tilde{B}\|^{-1} \quad (7.2)$$

where $\tilde{B} = \partial \tilde{Q} / \partial \tilde{Q}_v$ are the preconditioned advective Jacobians transformed to the master element.

As is well known, the computation of intrinsic time scale matrix is one of the most time-consuming parts of CFD codes. The numerical solution of the eigenvalue problem associated with the computation of some norm of the advective Jacobians is responsible for this high computational cost.

The most challenging applications nowadays requires the usage of vector and parallel machines that involve the development of specific software for such architectures. Optimized numerical routines are designed for very huge arrays of data that are divided among processors to do the computation faster. But in this case, the τ definition is based on the solution of a large amount of small eigenvalue problems (5×5 in 3D) and there are no optimal library routines to accomplish this. In this sense, it is better to develop some algebraic expressions that are easily vectorized or parallelized. A lot of effort has been dedicated to the development of equivalent ad-hoc expressions for this matrix that are designed only for some specific problems and each time that you face with a new problem you have to modify it according to the mathematical model. On the other hand, the above definition is general and can be directly used in every problem governed by a system of equations similar to (6.1).

In this paper we present an analytical approach for the computation of this time characteristic matrix avoiding the usage of time consuming numerical routines of eigenvalues and eigenvectors for its definition. For the sake of simplicity we transform back the problem to the viscous basis

$$\frac{\partial Q_v}{\partial t} + \tilde{A}_v \cdot \nabla Q_v = \tilde{K}_v \nabla^2 Q_v + \tilde{F}_v \quad \text{with} \quad \begin{cases} \tilde{A}_v = \Gamma_v^{-1} A \frac{\partial Q}{\partial Q_v} \\ \tilde{K}_v = \Gamma_v^{-1} K \frac{\partial Q}{\partial Q_v} \\ \tilde{F}_v = \Gamma_v^{-1} F \end{cases} \quad (8.1-4)$$

(see Appendix A for the expressions of some of these matrices).

So, the numerical diffusivity tensor introduced by SUPG is

$$\begin{aligned}\tilde{\mathbf{K}}_v^{\text{num}} &= \tilde{\mathbf{A}}_v \tilde{\tau}_v \tilde{\mathbf{A}}_v \\ &= \Gamma_v^{-1} \mathbf{A} \frac{\partial \mathbf{Q}}{\partial \mathbf{Q}_v} \tilde{\tau}_v \Gamma_v^{-1} \mathbf{A} \frac{\partial \mathbf{Q}}{\partial \mathbf{Q}_v}\end{aligned}\quad (9.1)$$

where

$$\tilde{\tau}_v = \|\tilde{\mathbf{B}}_v\|^{-1} \quad (9.2)$$

and

$$\tilde{\mathbf{B}}_{v_i} = \frac{\partial \xi_i}{\partial x_j} \tilde{\mathbf{A}}_{v_j} \quad (9.3)$$

Now, we transform back to the original system (6.1)

$$\frac{\partial \mathbf{Q}}{\partial t} + \tilde{\mathbf{A}} \cdot \nabla \mathbf{Q} = \left(\tilde{\mathbf{K}} + \tilde{\mathbf{A}} \frac{\partial \mathbf{Q}}{\partial \mathbf{Q}_v} \tilde{\tau}_v \Gamma_v^{-1} \mathbf{A} \right) \nabla^2 \mathbf{Q} + \tilde{\mathbf{F}} \quad (10)$$

with the new time characteristic matrix

$$\tilde{\tau} = \frac{\partial \mathbf{Q}}{\partial \mathbf{Q}_v} \tilde{\tau}_v \Gamma_v^{-1} \quad (11)$$

As is shown in the above expression, the new time characteristic matrix is based on the computation of

$$\begin{aligned}\tilde{\tau}_v^{-1} &= \|\tilde{\mathbf{B}}_v\| = \sum_i |\tilde{\mathbf{B}}_{v_i}| \\ &= \sum_i \left| \sum_j \frac{\partial \xi_i}{\partial x_j} \tilde{\mathbf{A}}_{v_j} \right| = \sum_i \left| \sum_j J_{ij} \tilde{\mathbf{A}}_{v_j} \right| = \sum_i |j_i \cdot \tilde{\mathbf{A}}_v|\end{aligned}\quad (12)$$

with $J_{ij} = \partial \xi_i / \partial x_j$ the coefficients of the Jacobian matrix between the physical domain and the master element. Each row of this 3×3 Jacobian represents a direction and we define it as j_i .

Then

$$\tilde{\tau}_v^{-1} = \sum_i |j_i \cdot \tilde{\mathbf{A}}_v| = \sum_i S_i |A_i| S_i^{-1} \quad (13)$$

need the computation of the eigenvalues A and eigenvectors S of the projection of $\tilde{\mathbf{A}}_v$ in each of the three directions j_i with $i = 1, 2, 3$. This problem can be solved explicitly without the requirement of numerical routines as in [25]

REMARK. As we will show later, the implementation of absorbent boundary conditions also need the solution of one eigensystem associated with the projection of the advective Jacobian along the outward boundary normal.

The eigenvalues are

$$A = \text{diag} \left(u, u, u, \frac{u(1 + \beta M^2 \omega / \gamma RT) \pm c'}{2} \right) \quad (14.1)$$

where $\omega = \gamma - (\gamma - 1)\delta$ and the pseudo-acoustic speed c' is defined as

$$c'^2 = u^2 \left(1 + \frac{\beta M^2 \omega}{\gamma RT} \right)^2 + 4\beta M^2 \left(1 - \frac{u^2}{\gamma RT} \right) \quad (14.2)$$

In this study we use $\delta = 1 \rightarrow \omega = 1$.

The computation of eigenvectors in matrix form

$$\tilde{\mathbf{A}}_v \cdot jS = SA \quad (15)$$

is equivalent to finding the null spaces of several linear systems, one for each column of S matrix,

The algebraic details and the eigenvectors are included in Appendix A.

SCIPEDIA

Register for free at <https://www.scipedia.com> to download the version without the watermark

It is important to remark that as was mentioned at the beginning of this section, this procedure is only valid for the inviscid case. The extension to Navier–Stokes equations should be accomplished by the correction of the $\tilde{\tau}$ matrix due to viscous effects. In this work we use an approximation to the *magic function* [17,24,26] to do it.

$$\tilde{\tau}^{\text{corr}} = \min\left(1, \frac{1}{9} \tilde{\text{Re}}^h\right) \tilde{\tau} \quad (16)$$

where $\tilde{\text{Re}}^h = (\|\mathbf{u}\|_h)/2\bar{\nu}$ is an element based Reynolds number modified by the preconditioner.

Here, we do not include any proof but it is our experience that the preconditioner does not change the physical meaning of the diffusive parameters keeping unaltered the Reynolds number. Then, the viscous correction can be achieved using the standard element Reynolds number.

Finally, the intrinsic time matrix included in the variational formulation (τ') should be defined from (11) and (16) taking into account the algorithm used to update the solution.

Then the whole procedure for $\tilde{\tau}$ computation is

ALGORITHM 1

Do $i = 1, 3$

1. Let Q be the solution at a gaussian point
2. Take \mathbf{n}^i the direction to project the advective matrices from the Jacobian of the transformation between physical and master element domain
3. Normalize $\hat{\mathbf{n}}^i = \mathbf{n}^i / \|\mathbf{n}^i\|$
4. $[\mathbf{A}^i, \mathbf{S}^i] = \text{call routine eigensystems}(\hat{\mathbf{n}}^i, Q)$
5. $\tilde{\tau}_v^{-1} = \tilde{\tau}_v^{-1} + \mathbf{S}^i [\mathbf{A}^i \mathbf{S}^{i-1}] \|\mathbf{n}^i\|$

end

$$6. \tilde{\tau}_v = (\tilde{\tau}_v^{-1})^{-1}$$

$$7. \tilde{\tau} = \frac{\partial Q}{\partial Q_v} \tilde{\tau}_v \Gamma_v^{-1}$$

$$8. \tilde{\tau}^{\text{corr}} \text{ from (16)}$$

end

Register for free at <https://www.scipedia.com> to download the version without the watermark

In the next section we present the definition of this matrix for the explicit and implicit cases.

5.2. Explicit implementation

In this section we show how to solve the preconditioned and semidiscretized SUPG formulation in a time marching fashion to get the steady solution. We use an explicit first-order forward Euler integration scheme

$$\frac{M}{\Delta t} \Delta Q = \tilde{R} \quad (17)$$

with $\tilde{R} = \Gamma^{-1} R$ and R the standard residual of the above formulation (3).

In order to keep the original non-preconditioned definition of the advective and diffusive Jacobians we have to modify the time intrinsic matrix in the SUPG term reaching a final expression like

$$\tau' = \tilde{\tau} \Gamma^{-1} \quad (18)$$

This explicit scheme contains a very strong restriction on the maximum time step allowable for getting stable results. The CFL condition controls the stability of this scheme.

In the case of highly refined non-uniform meshes the CFL condition represents a very severe restriction to reach the steady state. To improve the rate of convergence for steady solutions we can use a local time stepping that is equivalent to use a constant Courant number independent of the element size.

The preconditioner applied to the original equations affects the value of the critical time step through its influence over the maximum eigenvalue (see Eq. (14.1–2)).

In this paper we do not show any result using explicit solver but we have done some simulations using it and the performance was good.

5.3. Matrix-free GMRES implicit implementation

In this section we extend the usage of the preconditioner to implicit solvers, specifically GMRES.

Rewriting the system in conservative variables (Eq. (2)), it is easy to see that the intrinsic time matrix remains the same as in the explicit case and the only difference with the original system is the definition of the mass matrix.

Then, the implementation of this preconditioner in GMRES is straightforward involving the only change of the mass matrix. As it is normal in GMRES we use also the *nodal block diagonal* right pre preconditioner to scale the equations and the algorithm can be written as:

ALGORITHM 2

1. *Start*: Choose ΔQ_0 and a dimension m of the Krylov subspace.
Define an $(m+1) \times m$ matrix H_m and initialize all its entries $h_{i,j}$ to zero
 2. *Arnoldi process*:
 - (a) Compute $r_0 = R - \tilde{M} \Delta Q_0$, $\beta = \|r_0\|_2$ and $v_1 = r_0 / \beta$
 - (b) For $j = 1, \dots, m$ do
 - # (1) Compute $z_j = D^{-1} v_j$
 - # (2) Compute $w = \tilde{M} z_j$
 - # (3) For $i = 1, \dots, j$, do $\begin{cases} h_{i,j} = (w, v_i) \\ w = w - h_{i,j} v_i \end{cases}$
 - # (4) Compute $h_{j+1,j} = \|w\|_2$ and $v_{j+1} = w / h_{j+1,j}$
 - (c) Define $V_m = [v_1, \dots, v_m]$
 3. *Form the approximate solution*:
Compute $\Delta Q_m = \Delta Q_0 + D^{-1} V_m y_m$ where
 $y_m = \argmin_y \|\beta e_1 - H_m y\|_2$ and $e_1 = [1, 0, \dots, 0]^T$
 4. *Restart*: If satisfied stop, else set $\Delta Q_0 \leftarrow \Delta Q_m$ and go to 2.
- end

Register for free at <https://www.scipedia.com> to download the version without the watermark

$$\begin{aligned} \tilde{M} &= \int_{\Omega} W^h \cdot \frac{\Gamma W^h}{\Delta t} d\Omega + \int_{\Omega} W^h \cdot A_i^h \frac{\partial Q^h}{\partial x_i} d\Omega + \int_{\Omega} \frac{\partial W^h}{\partial x_i} \cdot K_{ij} \frac{\partial W^h}{\partial x_j} d\Omega \\ &+ \sum_{e=1}^{\text{numel}} \int_{\Omega^e} (\tau')^T (A_k^h)^T \frac{\partial W^h}{\partial x_k} \cdot \left\{ \frac{\Gamma W^h}{\Delta t} + A_i^h \frac{\partial W^h}{\partial x_i} - \frac{\partial}{\partial x_i} \left(K_{ij} \frac{\partial W^h}{\partial x_j} \right) \right\} d\Omega \\ &= \frac{\Gamma M}{\Delta t} + C(Q^{(n)}) \end{aligned} \quad (19)$$

and D = nodal block diagonal of (\tilde{M}) .

The Arnoldi process builds an orthogonal basis of the right preconditioned Krylov subspace

$$\text{Span}\{r_0, \tilde{M} D^{-1} r_0, \dots, (\tilde{M} D^{-1})^{m-1} r_0\} \quad (20)$$

The matrix-free GMRES version is accomplished by considering that

$$\tilde{M} \Delta Q = \frac{\Gamma M}{\Delta t} \Delta Q + C(Q^{(n)}) \Delta Q = \frac{\Gamma M}{\Delta t} \Delta Q - R(Q^{(n)}, \Delta Q) \quad (21)$$

where the residual has two arguments, the first one is the known state variable at the last time step that is frozen inside the GMRES loop and the second one is any other vector that performs like a perturbation around the local value $Q^{(n)}$ [1].

Then the whole algorithm is written as:

ALGORITHM 3
 Given $Q^{(n)}$,
 (1) Form $R(Q^{(n)})$
 (2) Form $D^{-1}(Q^{(n)})$ using Eq. (19)
 (3) Solve $\tilde{M} \Delta Q = R$ using Algorithm 2
 (4) Update $Q^{(n+1)} = Q^{(n)} + \Delta Q$
 end

5.4. Absorbent boundary condition

Another important aspect of the new preconditioned system is related to the implementation of the absorbent boundary condition.

This non-reflective boundary operator is defined by the solution of an eigenvalue problem for the advective Jacobian projected on the outward unit normal.

In the standard non-preconditioned problem the following steps are carried out:

ALGORITHM 4
 1. $A \cdot \hat{n} = S \Lambda S^{-1}$
 2. $V^{\text{int}} = S^{-1} Q^{\text{int}}$
 $V^{\text{ext}} = S^{-1} Q^{\text{ext}}$
 3. $V = \Pi_{\lambda \geq 0} V^{\text{int}} + \Pi_{\lambda < 0} V^{\text{ext}}$
 4. $Q = SV$
 end

where $\Pi_{\lambda \geq 0}$ represents the projection operator on the span generated by the non-negative eigenvalues and $\Pi_{\lambda < 0}$ is the same for the negative ones.

As can be seen in the above algorithm, we need to use the eigenvalue decomposition of the preconditioned advective Jacobian instead of the original one, but at the same time we do not need to compute the eigenvectors (see Algorithm 1) but using the outward normal unit vector instead of each Jacobian row.

6. Numerical results

The goal of this paper is to show the effectiveness of this physics based preconditioner to solve compressible and nearly incompressible viscous and inviscid flow problems using an SUPG formulation with a matrix-free version of GMRES implicit integration scheme. In this section we present four different types of numerical experiments:

- (1) High Reynolds number and inviscid flow on nearly incompressible flow. ($Re \rightarrow \infty$, $M \rightarrow 0$).
- (2) Behavior to several Reynolds number, specially at low values ($0 < Re < \infty$, $M \rightarrow 0$).
- (3) Comparison with standard formulation at subsonic regimes.
- (4) Extension to three-dimensional flow.

These experiments allow us to recognize the behavior of the preconditioner for problems with a wide range of Mach and Reynolds numbers inside the domain.

To gain some insight about the performance of the preconditioner presented here we solve some simple examples like flow around of a thin wing (NACA0012) and flow around of a circular cylinder and a sphere for several Reynolds and Mach numbers.

The mesh employed for the NACA0012 simulation is shown in Fig. 1 and it consists of a C-grid with 39 nodes on the airfoil and 20 along the wake and 31 nodes in the normal to the streamwise direction. We have used our 3D code using one slab of elements in the third dimension and periodical boundary condition to eliminate spurious perturbations in this direction. The angle of attack was always zero degree and we mesh the

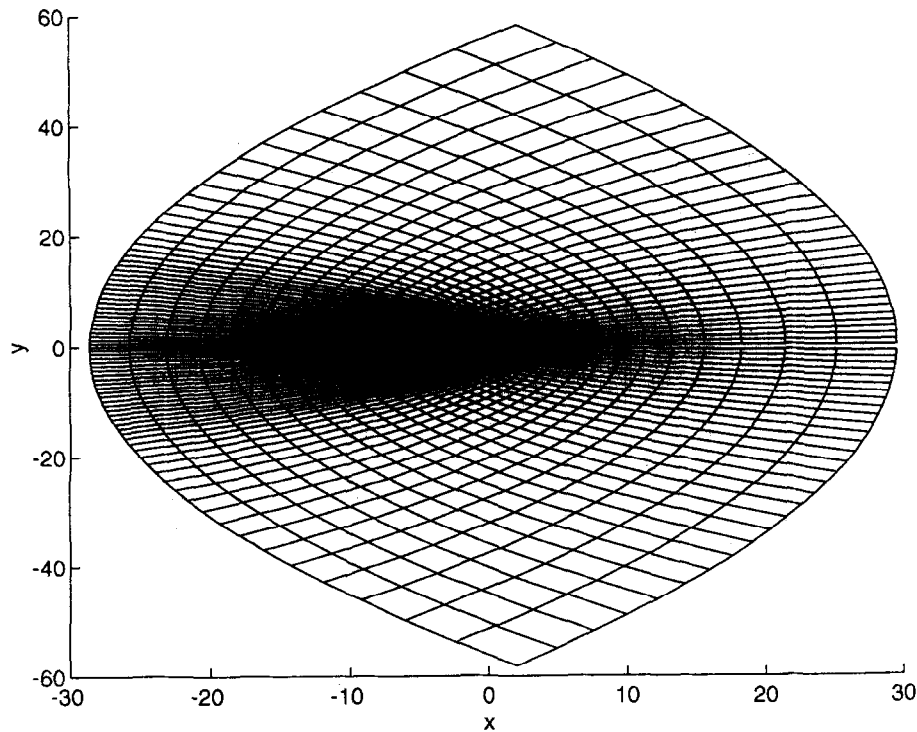


Fig. 1. NACA0012—Mesh.

lower and the upper side of the airfoil. The chord size of the airfoil is 2 units and the far field boundary is placed at more than 28 units. The boundary conditions for Euler computation were slip velocity on the airfoil, absorbent boundary condition on the far field and we use the Kutta condition on the trailing edge. For Navier–Stokes we change the slip and the Kutta condition for non-slip velocities and null heat flux along the profile.

For the circular cylinder example we mesh the entire domain using 2 zones, the first close to the body is built with an O-mesh with 80 nodes around the cylinder and 20 in the radial direction and the second zone is an H-mesh with 20 nodes in the downstream direction (see Fig. 2). The diameter of the cylinder is 1 unit and we have placed the inlet, upper and lower boundary at 9 units and the outlet boundary at 27 units. The boundary conditions were non-slip around the cylinder with null heat flux and we have imposed all the variables at the inlet, upper and lower boundaries while using free traction and null heat flux at the outlet.

For the sphere we have used 19 942 nodes and 15 840 elements distributed inside a computational domain, a box with $-3 \leq x \leq 10$, $-3 \leq y \leq 3$, $-3 \leq z \leq 3$ and a sphere diameter equal to one. Boundary conditions were similar to the circular cylinder case.

The computational cost associated with the preconditioner was tested for all the examples presented here and our measurements lead to an increment of 30% in the CPU-time.

6.1. High Reynolds number and inviscid flow on nearly incompressible flow

Case 1A: NACA 0012	Ma = 0.001	Euler
--------------------	------------	-------

Fig. 3 shows the convergence of the state variables ΔQ (left) and the residual (right) for the inviscid case. We can see in the left plot that all the equations have the same rate of convergence (same slope) showing that the condition of the system is strongly improved by the preconditioner. Remember that for this example in the non-preconditioned system the acoustic waves travel 1000 times faster than the particle waves and the slope should differ in this factor. The right plot shows that the residual converges with almost the same rate and after

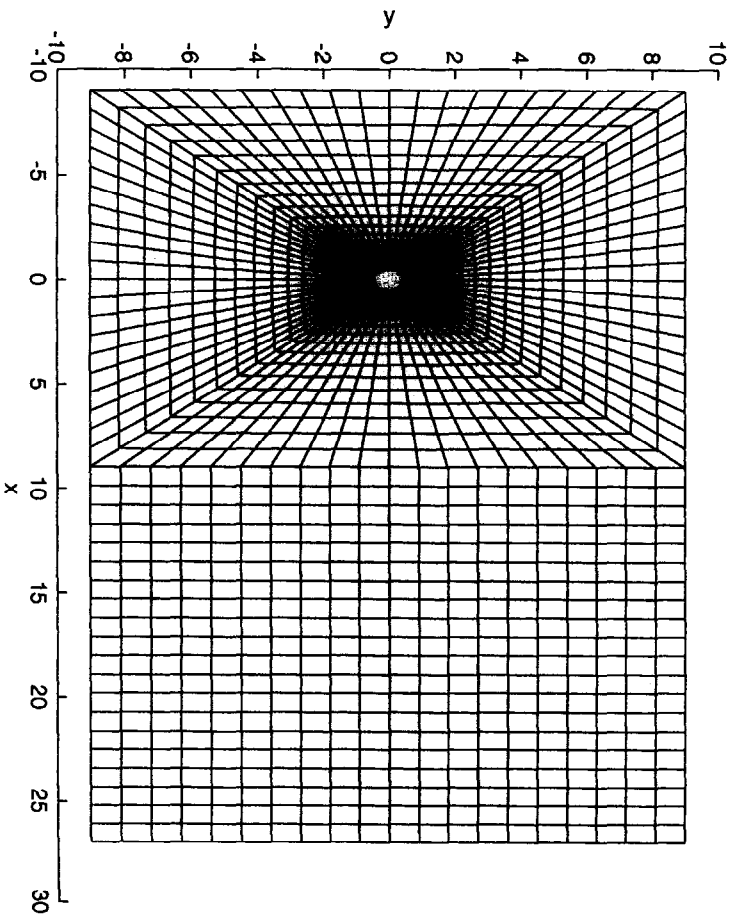
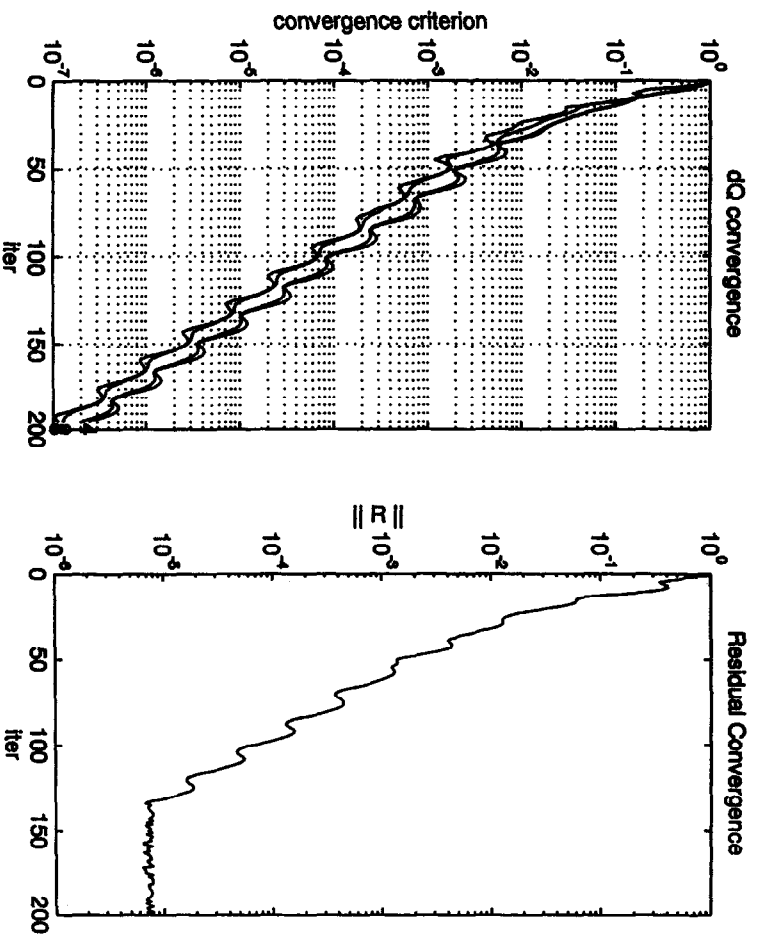


Fig. 2. Circular cylinder.

Fig. 3. NACA 12—Euler— $Ma = 0.001$ —GMRES & phys preco—Convergence.

130 iterations the system is dominated by round-off errors and the residual stagnates in 10^{-5} times the original residual. This difficulty can be solved using a redefined gauge pressure instead of the standard one for the computation of the pressure gradients. This kind of strategy is not presented here and it is under study. The non-preconditioned results are not presented because for such a low Mach number the system becomes numerically unstable with checkerboard modes in the pressure field and it does not make sense of its comparison. Fig. 4 shows the pressure coefficient (left) and the total pressure coefficient (right) for the preconditioned solution. We note that even though coarse mesh was employed in this computation we reach total pressure values close to one, the exact value of inviscid flow.

Case 1B: NACA 0012 $Ma = 0.001$ $Re = 200$
--

Fig. 5 shows the convergence in the equations (left) and the residual (upper right) for the same airfoil but now solving Navier–Stokes equations with a relatively high Reynolds number. We can note that the good behavior of the convergence rate shown in the inviscid case is not lost. Moreover, the solution continues being numerically stable (lower right plot) in contrast with the non-preconditioned solution that exhibits numerical oscillations.

6.2. Reynolds number sensibility for nearly incompressible flow

Case 2A: NACA 0012 $Ma = 0.001$ $Re = 1$
--

Fig. 6 shows the convergence in ΔQ and the convergence in the residual for a low Reynolds viscous flow through a NACA0012 airfoil. These numerical experiments allow us to understand the behavior of the preconditioner for extremely viscous flows. This is a very good test because it contains a mechanism of error reduction completely different to that presented in the first two examples. Here, the dissipation of the error is very important and in general viscous effects compete strongly with acoustic waves error transportation. It is deduced from the left plot that all the error waves are damped at almost the same rate.

Case 2B: NACA 0012 $Ma = 0.001$ $Re = 400$
--

Fig. 7 shows the same as Fig. 6 but now for $Re = 400$. The conclusions about the convergence are equivalent to Case 1B ($Re = 200$).

In Fig. 8 we plot the results for the NACA0012 airfoil case altogether. We note that the inviscid case shows the best convergence rate, approximately twice the viscous cases that are almost equivalent among them.

Case 2C: Circular cylinder $Ma = 0.001$ $Re = 20$

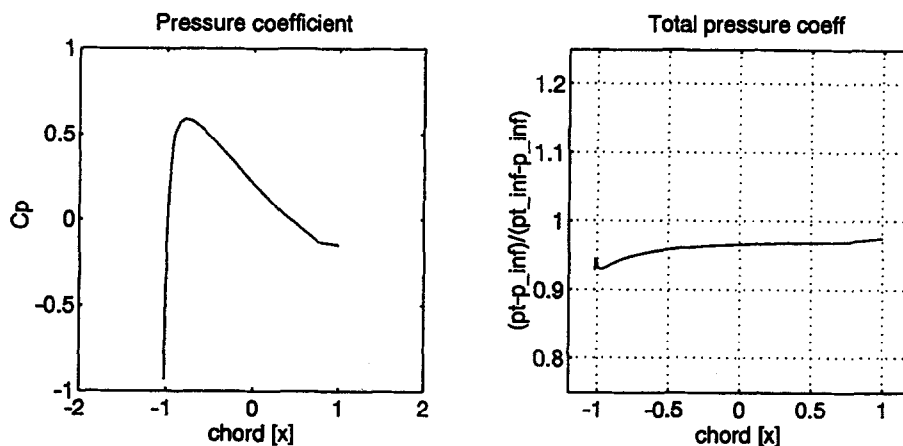
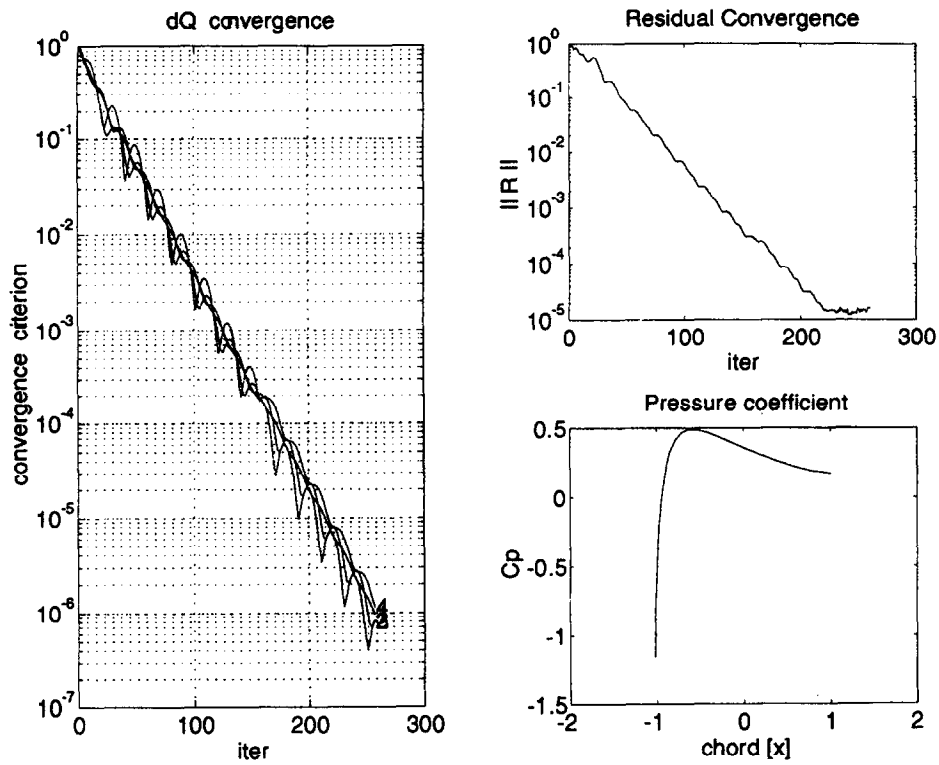
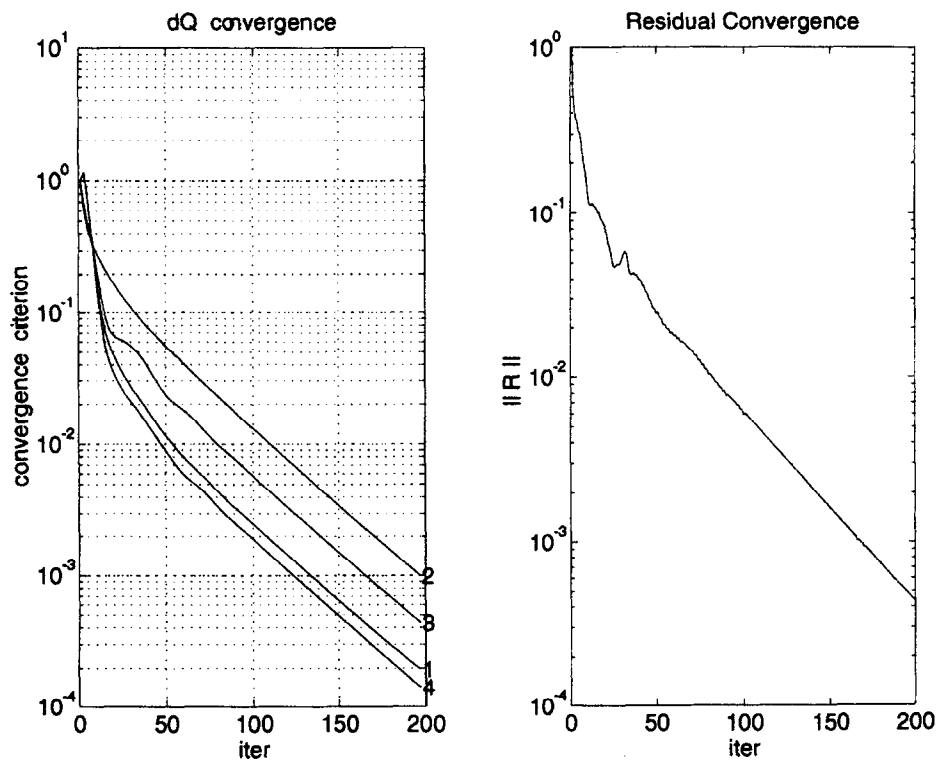


Fig. 4. NACA 12—Euler— $Ma = 0.001$ —GMRES & phys preco. Pressure coefficients.

Fig. 5. NACA 12— $Ma = 0.001$ — $Re = 1$ —GMRES & phys preco.Fig. 6. NACA 12— $Ma = 0.001$ — $Re = 400$ —GMRES & phys preco.

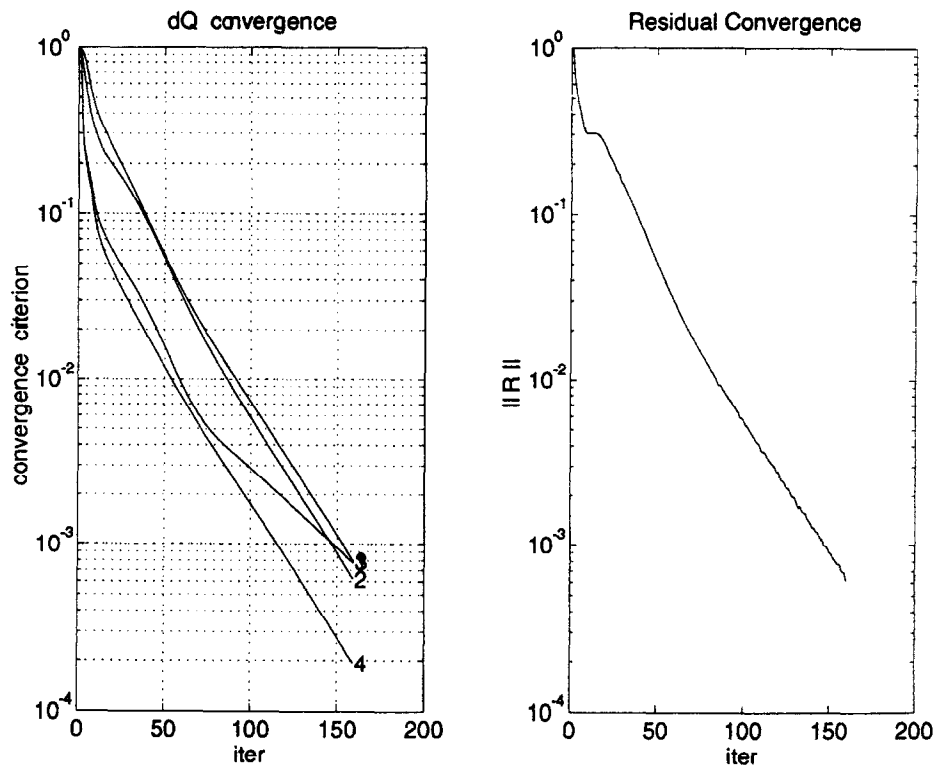


Fig. 7. NACA—Ma = 0.001—Re = 400—GMRES & phys preco.

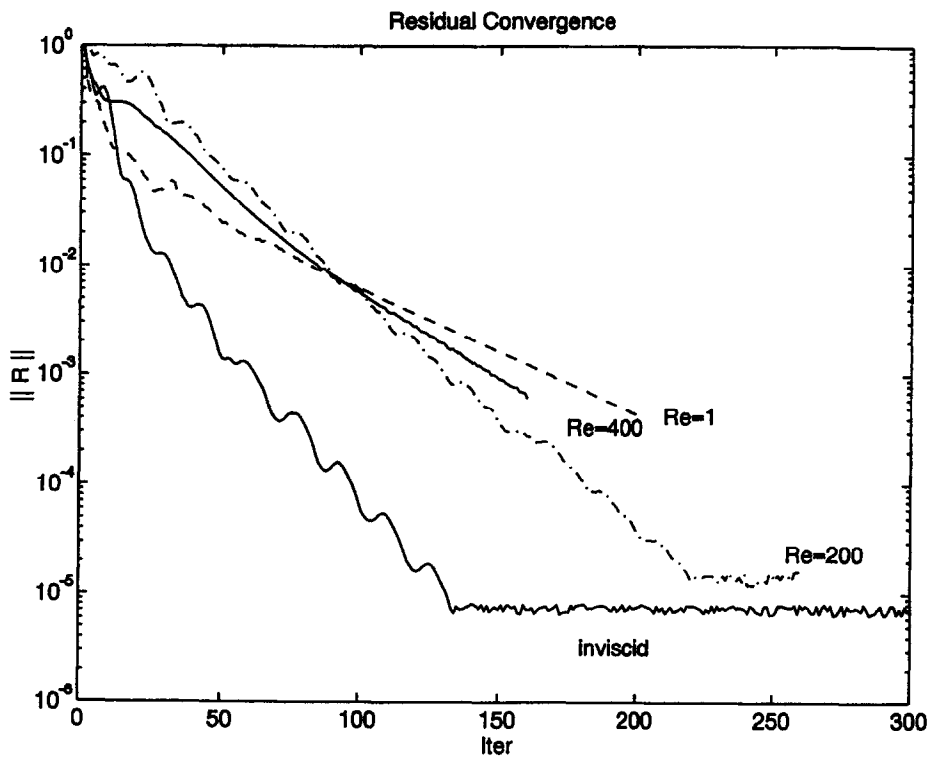


Fig. 8. NACA 12—GMRES & phys preco—Convergence.

Fig. 9 show the convergence reached when we run the circular cylinder with $Re = 20$ based on the diameter of the cylinder. The residual and the ΔQ go down without deterioration

Case 2D: Circular cylinder	$Ma = 0.001$	$Re = 1$
----------------------------	--------------	----------

Fig. 10 is similar to Fig. 9 but using $Re = 1$. The residual has almost the same behavior as the last case (right plot) but we can note that the continuity equation (left plot) has a different slope respect to the others. This phenomenon is under study but it does not influence the residual convergence. Several factors like strong viscous effects, the coarseness of the mesh and the round-off errors could be affecting this slope.

6.3. Comparison with standard formulation at subsonic regimes

In this third kind of result we present a comparison between the preconditioned system and the original standard non-preconditioned one in order to partially demonstrate that this preconditioner does work for all Mach numbers. Here, we present a flow around the NACA0012 airfoil for a subsonic condition.

Case 3: NACA 0012	$Ma = 0.3$	$Re = 200$
-------------------	------------	------------

Fig. 11 shows the results for the preconditioned scheme and Fig. 12 the same for the standard scheme. Fig. 13 presents a comparison between the residual convergence for both solutions. Both have almost the same slope and we can guess that this behavior will keep for all the Mach numbers. But, we need further research in order to strengthen this conclusion.

6.4. Extension to three-dimensional problems

We choose the sphere as a three-dimensional example to show the performance of the preconditioner to solve this kind of problem. 3D problems are very useful examples to test the numerical implementation of the preconditioner concerned with the stabilization of the scheme. Flow around the sphere is a very interesting problem because it has an analytical solution for Stokes flow and some approximations for low Reynolds numbers. Also, there is some correlation between the Reynolds number and the extension of the recirculation pocket in the rear part of the body.

Case 4A: Sphere	$Ma = 0.002$	$Re = 10$
-----------------	--------------	-----------

Fig. 14 shows the convergence history and Fig. 15 the streamlines for flow around a sphere at low Reynolds number. The convergence seems to be almost optimal in the sense that all the equations converge with the same rate. Only the continuity equation differs from the rest in a very small amount and the residual goes down six orders of magnitude in 100 GMRES iterations with a constant rate. After 100 iterations some round-off errors dominate and the convergence stagnates. For this value of Reynolds number there is more circulation at the rear part of the sphere and Fig. 15 confirms this pattern.

Case 4B: Sphere	$Ma = 0.002$	$Re = 100$
-----------------	--------------	------------

In this example we increase the Reynolds number to 100 and Fig. 16 shows that the convergence rate is good but the continuity equation convergence is worse than the other case. However, the residual convergence is not deteriorated by this phenomenon, similar behavior to the example 2D. Fig. 17 plots the streamlines for the plane

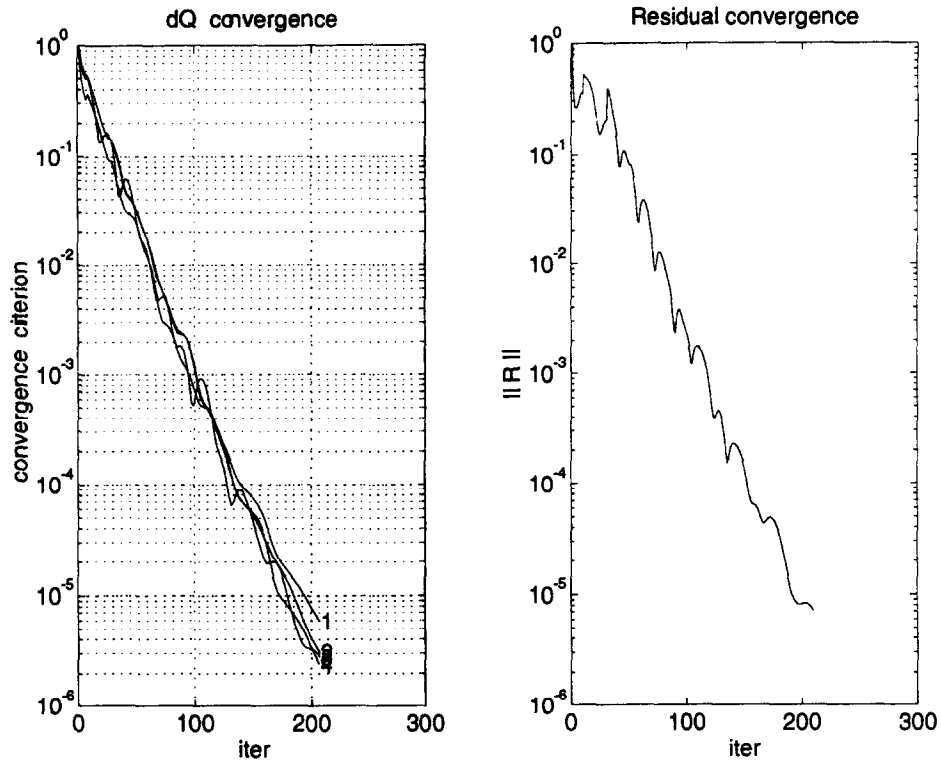


Fig. 9. Circular cylinder— $Ma = 0.001$ — $Re = 20$ —GMRES & phys preco.

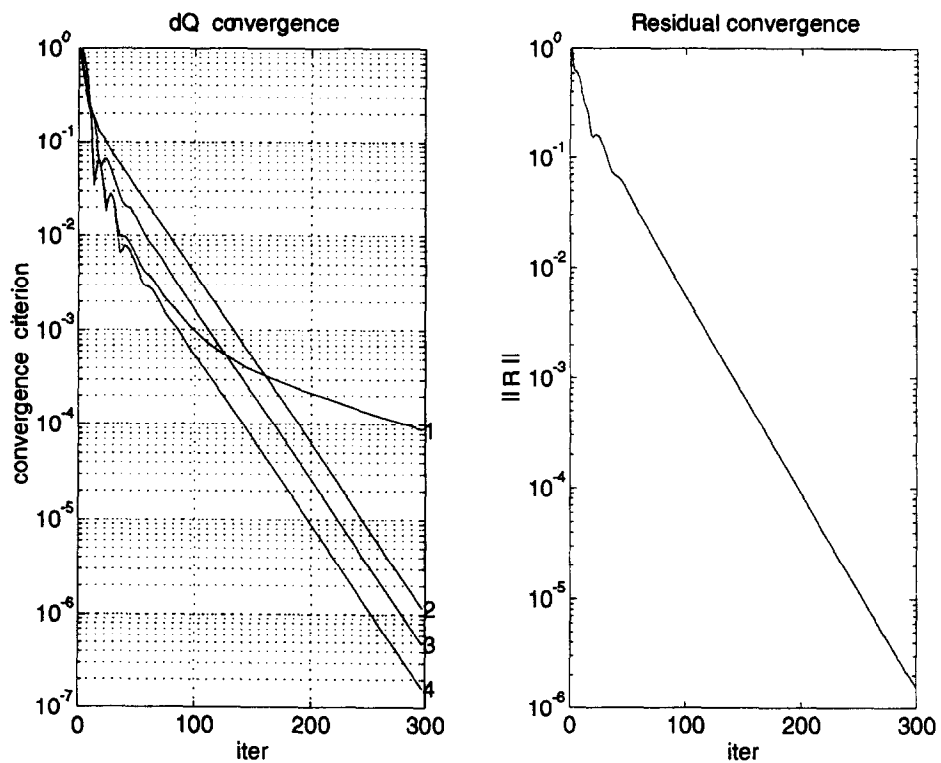


Fig. 10. Circular cylinder— $Ma = 0.001$ — $Re = 1$ —GMRES & phys preco.

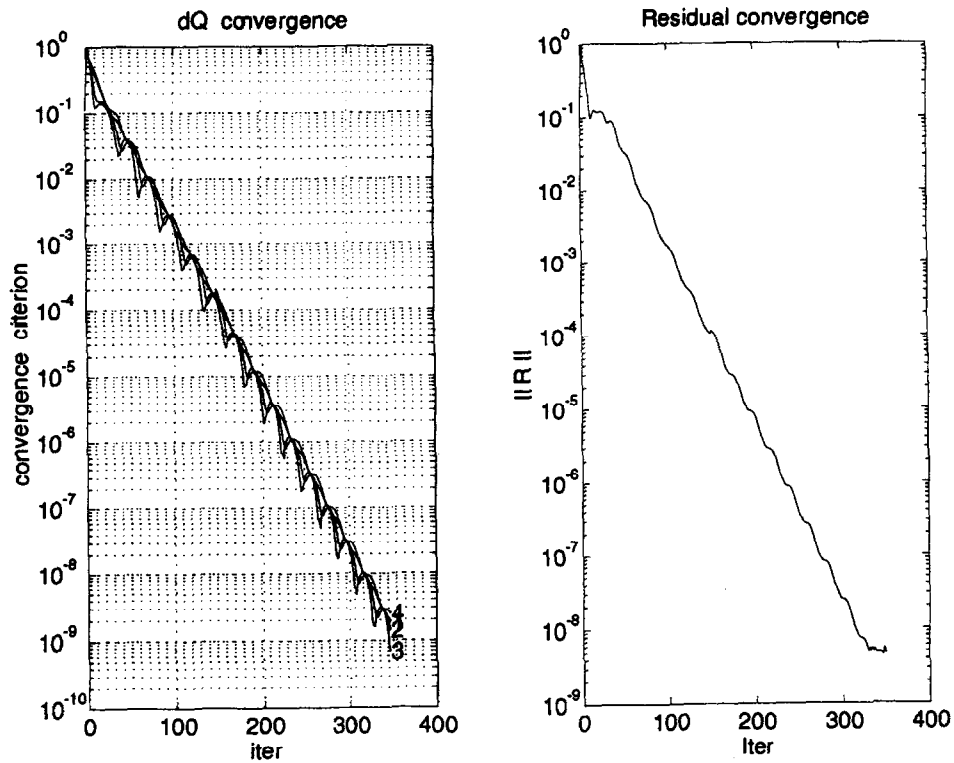


Fig. 11. NACA 12—Ma = 0.3—Re = 200—GMRES & phys preco.

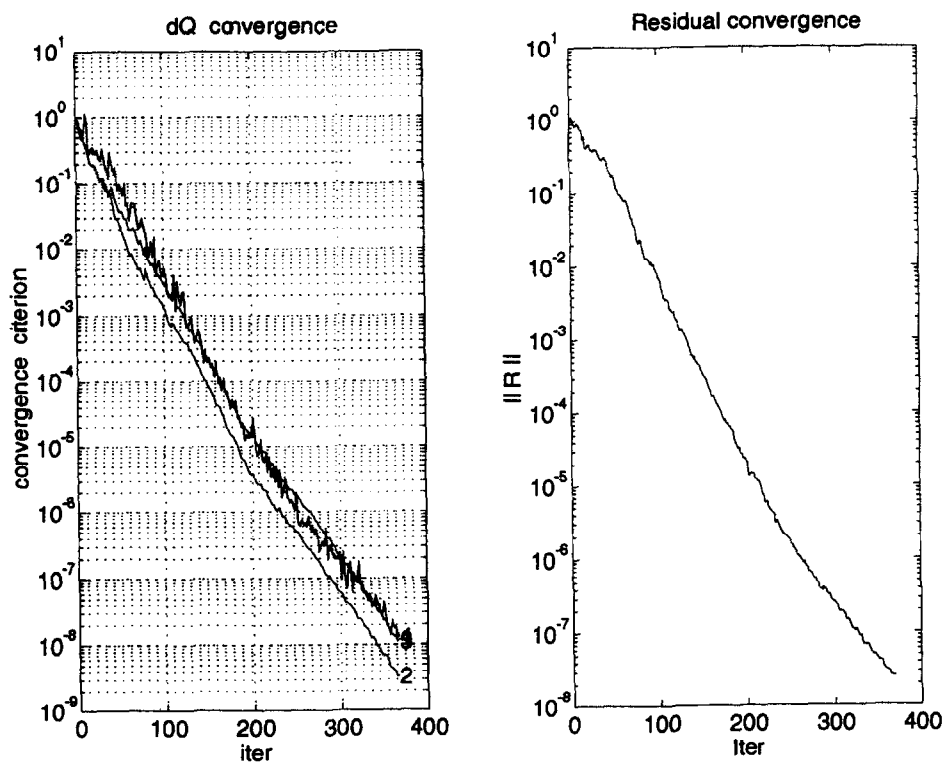


Fig. 12. NACA—Ma = 0.3—Re = 200—Std GMRES.

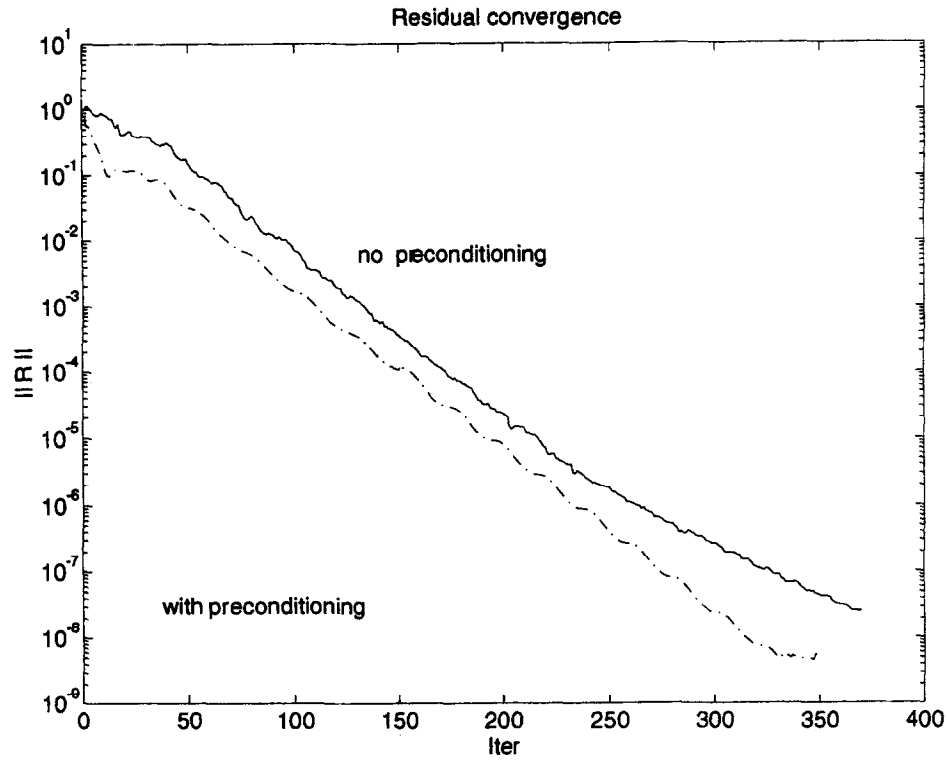


Fig. 13. NACA 12— $Ma = 0.3$ — $Re = 200$ —Comparison.

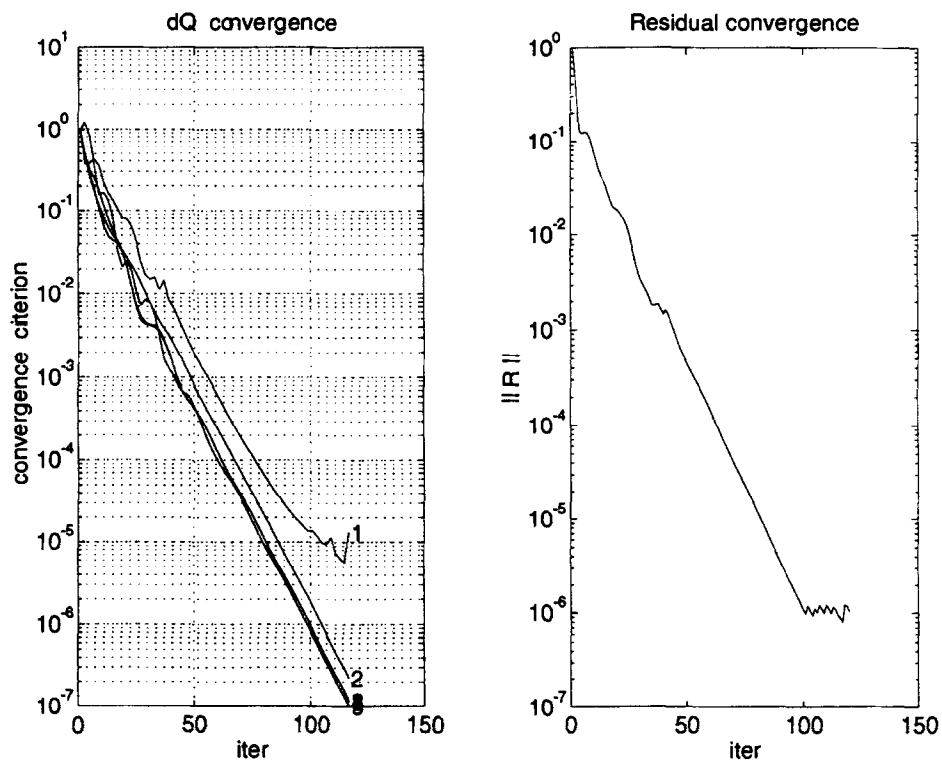
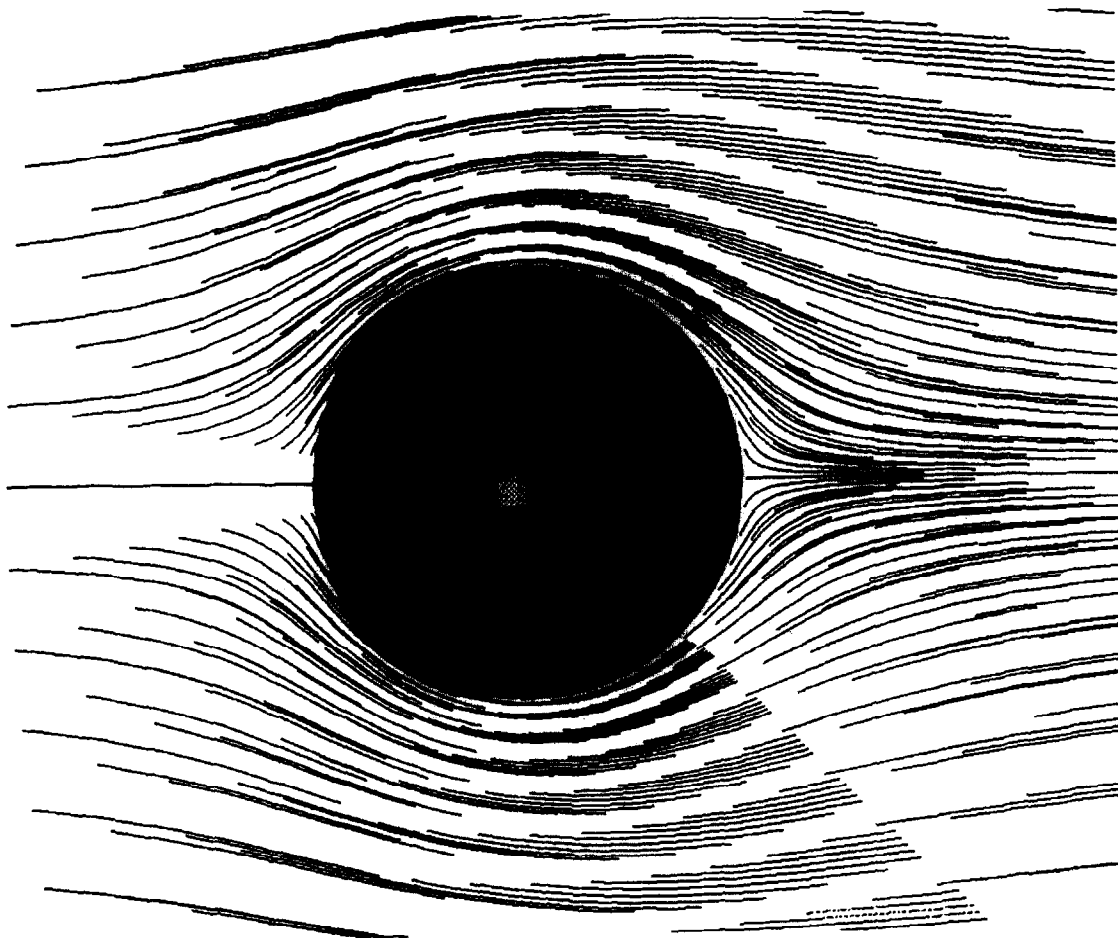
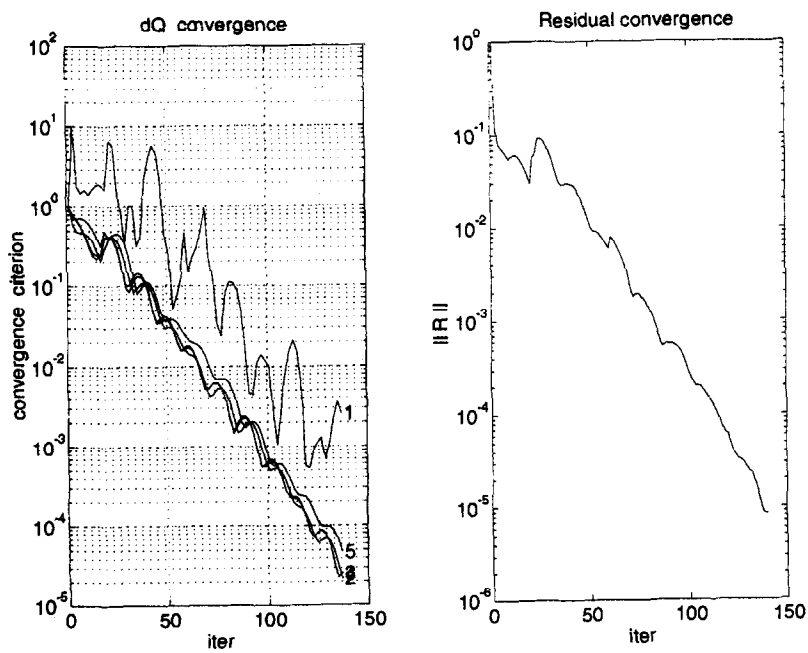


Fig. 14. Sphere— $Ma = 0.002$ — $Re = 10$.

Fig. 15. Sphere—Streamlines— $Re = 10$ — $Mach = 0.002$.Fig. 16. Sphere— $Ma = 0.002$ — $Re = 100$.

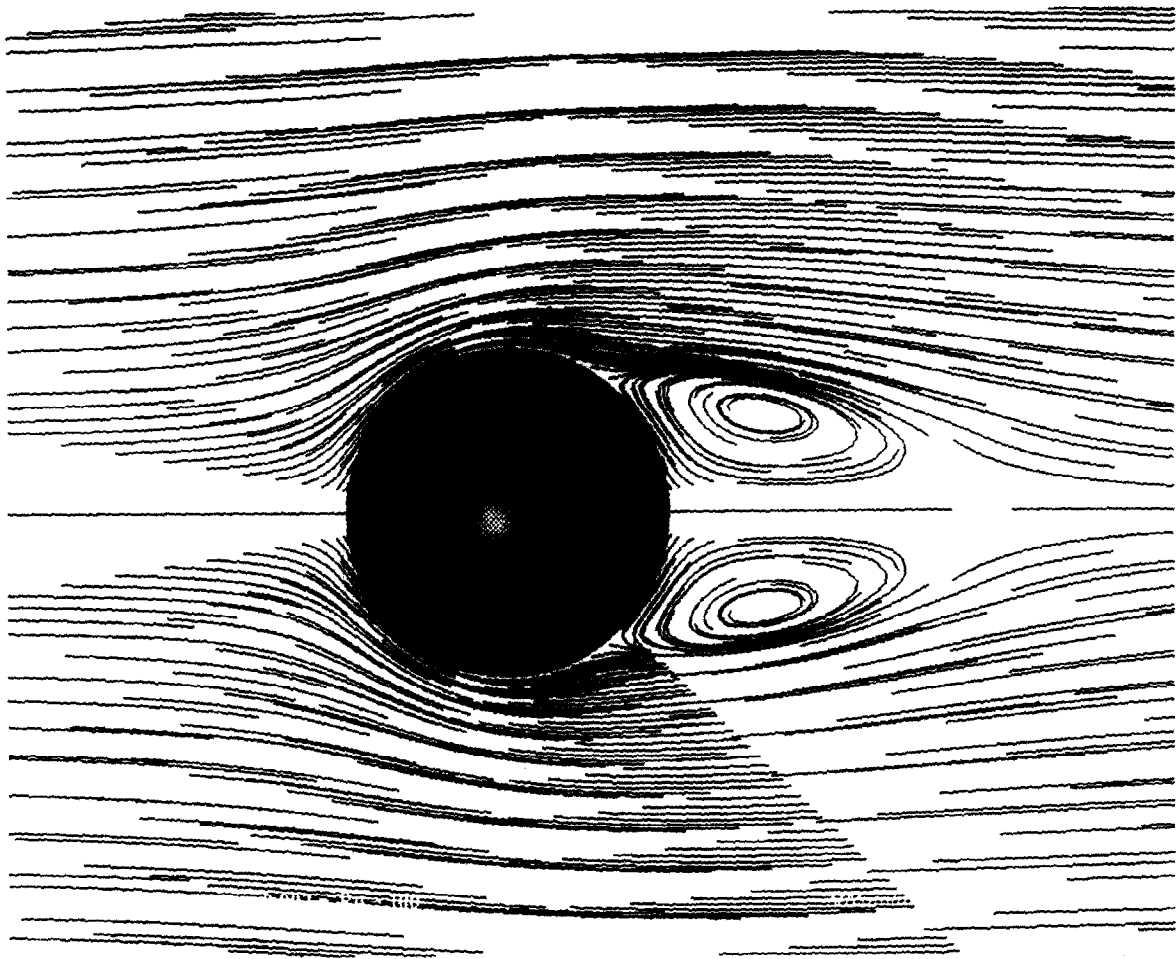


Fig. 17. Sphere—Streamlines— $Re = 100$ — $Mach = 0.002$.

$z = 0$ and the recirculation zone at the rear part of the sphere has an extension that is in a good agreement with experiments and some analytical approximations.

7. Conclusions

This paper has been a first trial to get an efficient implementation of an unified approach to compressible and incompressible flow. Also, it has shown to work quite good as a preconditioner for GMRES implicit solver and its local character makes it very suitable for future parallel implementations. Future research will be directed towards the application of this method to solve more complex fluid flow, for example helicopter and turbomachinery rotors. Another problem that is under study is the influence of high aspect ratio meshes on the condition number and the usage of this preconditioner to improve the convergence rate in these situations. Our goal is oriented to extend the effectiveness of the preconditioner presented in this paper when some turbulence model is used.

Acknowledgement

The first author wish to express his gratitude to Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET, Argentina) for its financial support. This work was sponsored in part by US Army, Army Research Laboratory, Army HPC Research Center. No official endorsement should be inferred. Cray C90 time was provided in part by the University of Minnesota Supercomputer Institute.

Appendix A

In this appendix we include a survey of the most important matrices employed in this work and some details about the computation of the eigensystem associated with the stabilizing terms for the preconditioner presented in this paper.

Transformation from viscous to conservative basis

$$\frac{\partial \mathbf{Q}_v}{\partial \mathbf{Q}} = \begin{pmatrix} \frac{\gamma-1}{2} \|\mathbf{u}\|^2 & -(\gamma-1)u & -(\gamma-1)v & -(\gamma-1)w & \gamma-1 \\ -\frac{u}{\rho} & \frac{1}{\rho} & 0 & 0 & 0 \\ -\frac{v}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 \\ -\frac{w}{\rho} & 0 & 0 & \frac{1}{\rho} & 0 \\ -\frac{T}{\rho} + \frac{1}{2} \frac{\|\mathbf{u}\|^2}{C_v \rho} & -\frac{u}{\rho C_v} & -\frac{v}{\rho C_v} & -\frac{w}{\rho C_v} & \frac{1}{\rho C_v} \end{pmatrix} \quad (\text{A.1})$$

Transformation from conservative to viscous basis

$$\frac{\partial \mathbf{Q}}{\partial \mathbf{Q}_v} = \begin{pmatrix} \frac{1}{(\gamma-1)C_v T} & 0 & 0 & 0 & -\frac{\rho}{T} \\ \frac{u}{(\gamma-1)C_v T} & \rho & 0 & 0 & -u \frac{\rho}{T} \\ \frac{v}{(\gamma-1)C_v T} & 0 & \rho & 0 & -v \frac{\rho}{T} \\ \frac{w}{(\gamma-1)C_v T} & 0 & 0 & \rho & -w \frac{\rho}{T} \\ \frac{1}{\gamma-1} + \frac{1}{2} \frac{\|\mathbf{u}\|^2}{(\gamma-1)C_v T} & \rho u & \rho v & \rho w & -\frac{1}{2} \|\mathbf{u}\|^2 \frac{\rho}{T} \end{pmatrix} \quad (\text{A.2})$$

Preconditioner in conservative basis

$$\mathbf{\Gamma} = \begin{pmatrix} \frac{\psi \|\mathbf{u}\|^2}{2} & -\psi u & -\psi v & -\psi w & \psi \\ \frac{1}{2} (\psi \|\mathbf{u}\|^2 - 2)u & 1 - \psi u^2 & -\psi uv & -\psi uw & \psi u \\ \frac{1}{2} (\psi \|\mathbf{u}\|^2 - 2)v & -\psi uv & 1 - \psi v^2 & -\psi vw & \psi v \\ \frac{1}{2} (\psi \|\mathbf{u}\|^2 - 2)w & -\psi uw & -\psi vw & 1 - \psi w^2 & \psi w \\ \Theta_1 & \Theta_2 u & \Theta_2 v & \Theta_2 w & 1 - \Theta_2 \end{pmatrix} \quad (\text{A.3.1})$$

where

$$\begin{aligned} \psi &= \frac{(\gamma-1)}{\beta M^2} \\ \Theta_1 &= -\frac{1}{2} \|\mathbf{u}\|^2 (1 + \Theta_2) - \frac{c^2}{\gamma-1} \\ \Theta_2 &= (1-\gamma)(1 + \Xi) \\ \Xi &= \frac{\rho e + p}{\rho \beta M^2} - \delta \end{aligned} \quad (\text{A.3.2-5})$$

and its inverse

$$\mathbf{I}^{-1} = \begin{pmatrix} \kappa & \frac{u}{C_v \gamma T} & \frac{v}{C_v \gamma T} & \frac{w}{C_v \gamma T} & \frac{-1}{C_v \gamma T} \\ (\kappa - 1)u & 1 + \frac{u^2}{C_v \gamma T} & \frac{uv}{C_v \gamma T} & \frac{uw}{C_v \gamma T} & \frac{-u}{C_v \gamma T} \\ (\kappa - 1)v & \frac{uv}{C_v \gamma T} & 1 + \frac{v^2}{C_v \gamma T} & \frac{vw}{C_v \gamma T} & \frac{-v}{C_v \gamma T} \\ (\kappa - 1)w & \frac{uw}{C_v \gamma T} & \frac{vw}{C_v \gamma T} & 1 + \frac{w^2}{C_v \gamma T} & \frac{-w}{C_v \gamma T} \\ \frac{1}{\psi} + \|\mathbf{u}\|^2 \left(\frac{1}{2} \kappa - 1 \right) & u \left(1 + \frac{1}{2} \frac{\|\mathbf{u}\|^2}{C_v \gamma T} \right) & v \left(1 + \frac{1}{2} \frac{\|\mathbf{u}\|^2}{C_v \gamma T} \right) & w \left(1 + \frac{1}{2} \frac{\|\mathbf{u}\|^2}{C_v \gamma T} \right) & -\frac{1}{2} \frac{\|\mathbf{u}\|^2}{C_v \gamma T} \end{pmatrix}$$

where

$$\kappa = \frac{\gamma \beta M^2 - (\gamma - 1) \|\mathbf{u}\|^2 + \beta M^2 \Xi (\gamma - 1)}{C_v \gamma (\gamma - 1) T} \quad (\text{A.4})$$

preconditioned viscous advective Jacobians

$$\tilde{\mathbf{A}}_{v_j} = \begin{pmatrix} \frac{u_j \beta M^2}{C_v T (\gamma - 1)} & \delta_{1j} \rho \beta M^2 & \delta_{2j} \rho \beta M^2 & \delta_{3j} \rho \beta M^2 & -\frac{\rho u_j}{T} \beta M^2 \\ \delta_{1j} \frac{1}{\rho} & u_j & 0 & 0 & 0 \\ \delta_{2j} \frac{1}{\rho} & 0 & u_j & 0 & 0 \\ \delta_{3j} \frac{1}{\rho} & 0 & 0 & u_j & 0 \\ \frac{u_j}{\rho (\gamma - 1) C_v} \left(\frac{1}{\rho} - \theta_5 \right) & \delta_{1j} T (1 - \theta_5) & \delta_{2j} T (1 - \theta_5) & \delta_{3j} T (1 - \theta_5) & u_j \theta_5 \end{pmatrix}$$

with

$$\theta_5 = \frac{\Xi \beta M^2 - \frac{1}{2} \|\mathbf{u}\|^2}{C_v \gamma T} \quad (\text{A.5})$$

projected preconditioned viscous advective Jacobian

$$\tilde{\mathbf{A}}_v \cdot \hat{\mathbf{n}} = \begin{pmatrix} \frac{u_n \beta M^2}{C_v T (\gamma - 1)} & \rho \beta M^2 n_x & \rho \beta M^2 n_y & \rho \beta M^2 n_z & -\frac{\rho u_n}{T} \beta M^2 \\ \frac{1}{\rho} n_x & u_n & 0 & 0 & 0 \\ \frac{1}{\rho} n_y & 0 & u_n & 0 & 0 \\ \frac{1}{\rho} n_z & 0 & 0 & u_n & 0 \\ \frac{u_n}{\rho (\gamma - 1) C_v} \left(\frac{1}{\rho} - \theta_5 \right) & T (1 - \theta_5) n_x & T (1 - \theta_5) n_y & T (1 - \theta_5) n_z & u_n \theta_5 \end{pmatrix} \quad (\text{A.6})$$

Eigenvectors

As we have presented in Section 5.1 we need to solve an eigensystem represented by

$$(\tilde{\mathbf{A}}_{v_n} - \lambda) \mathbf{S} = \mathbf{0}$$

where $\tilde{\mathbf{A}}_{v_n} = \tilde{\mathbf{A}}_v \cdot \hat{\mathbf{n}}$.

For each eigenvector (column of S) we have the following defective linear system to solve

$$\begin{aligned}
 & \left(\frac{1}{C_v T \psi} u_n - \lambda_i \right) s_1^i + \rho \beta M^2 (n_x s_2^i + n_y s_3^i + n_z s_4^i) - \frac{\rho \beta M^2 u_n}{T} s_5^i = 0 \\
 & \frac{n_x}{\rho} s_1^i + (u_n - \lambda_i) s_2^i = 0 \\
 & \frac{n_y}{\rho} s_1^i + (u_n - \lambda_i) s_3^i = 0 \\
 & \frac{n_z}{\rho} s_1^i + (u_n - \lambda_i) s_4^i = 0 \\
 & \frac{\left(1/\gamma - \frac{\Xi \beta M^2 - \frac{1}{2} \|u\|^2}{C_v \gamma T} \right) \frac{1}{C_v T \psi} T u_n}{\rho \beta M^2} s_1^i \\
 & + T \left(1 - \frac{\Xi \beta M^2 - \frac{1}{2} \|u\|^2}{C_v \gamma T} \right) ((n_x s_2^i + n_y s_3^i + n_z s_4^i) + \left(\frac{\Xi \beta M^2 - \frac{1}{2} \|u\|^2}{C_v \gamma T} u_n - \lambda_i \right) s_5^i) = 0
 \end{aligned} \tag{A.7.1-5}$$

where $i = 1, \dots, 5$ means each eigenvector (column of S).

For $\lambda_{1,2,3} = u_n$ we have

$$S(:, 1:3) = \begin{pmatrix} 0 & 0 & 0 \\ x^2 + \frac{u_n}{T} x^1 & x^3 & \frac{u_n}{T} x^1 \\ 1 & 0 & 1 \end{pmatrix} \tag{A.8}$$

where a set of three orthogonal vectors x^1, x^2, x^3 is chosen by the following procedure:

<i>Procedure</i>	
1.	$[x_1^1; x_2^1; x_3^1] = [n_x; n_y; n_z]$
2.	if $n_x = 0$ then
	$x_1^2 = n_x$
	$x_2^2 = -n_z$
	$x_3^2 = n_y$
	elseif $n_y = 0$ then
	$x_1^2 = -n_z$
	$x_2^2 = n_y$
	$x_3^2 = n_x$
	elseif $n_z = 0$ then
	$x_1^2 = -n_y$
	$x_2^2 = n_x$
	$x_3^2 = n_z$
	else
	$[x_1^2; x_2^2; x_3^2] = [x_1^1; x_2^1; x_3^1] \times [1; 0; 0]$
	endif
3.	$[x_1^3; x_2^3; x_3^3] = [x_1^1; x_2^1; x_3^1] \times [x_1^2; x_2^2; x_3^2]$
4.	$[x_1^3; x_2^3; x_3^3] = [x_1^3; x_2^3; x_3^3] \times [x_1^1; x_2^1; x_3^1]$
	<i>end</i>

For $\lambda_{4,5} = \frac{1}{2}((\kappa + \alpha/c^2)u_n \pm c')$ we have

$$S(:, 4:5) = \begin{pmatrix} \frac{1}{n_x} & \frac{1}{n_x} \\ -\frac{\rho(u_n - \lambda_4)}{n_y} & -\frac{\rho(u_n - \lambda_5)}{n_y} \\ \frac{\rho(u_n - \lambda_4)}{n_z} & \frac{\rho(u_n - \lambda_5)}{n_z} \\ -\frac{\rho(u_n - \lambda_4)}{s_5^4} & -\frac{\rho(u_n - \lambda_5)}{s_5^5} \end{pmatrix} \quad (\text{A.9})$$

with s_5^4 and s_5^5 from A.7.5.

References

- [1] S. Aliabadi, Parallel finite element computations in aerospace applications, Ph.D. Thesis, Department of Aerospace Engineering and Mechanics, University of Minnesota, 1994.
- [2] S. Aliabadi, S. Ray and T. Tezduyar, SUPG finite element computation of viscous compressible flows based on the conservation and entropy variables formulations, *Comput. Mech.* 11 (1993) 300–312.
- [3] S. Aliabadi and T. Tezduyar, Space-time finite element computation of compressible flows involving moving boundaries and interfaces, *Comput. Methods Appl. Mech. Engrg.* 107 (1993) 209–223.
- [4] C. Baumann, M. Storti and S. Idelsohn, A Petrov–Galerkin technique for the solution of transonic and supersonic flows, *Comput. Methods Appl. Mech. Engrg.* 95 (1992) 49–70.
- [5] M. Behr, A. Johnson, J. Kennedy, S. Mittal and T. Tezduyar, Computation of incompressible flows with implicit finite element implementations on the Connection Machine, *Comput. Methods Appl. Mech. Engrg.* 108 (1993) 99–118.
- [6] M. Behr and T. Tezduyar, Finite element solution strategies for large-scale flow simulations, *Comput. Methods Appl. Mech. Engrg.* 112 (1994) 3–24.
- [7] A. Brooks and T. Hughes, Streamline upwind Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, *Comput. Methods Appl. Mech. Engrg.* 32 (1982) 199–259.
- [8] Y. Choi and C. Merkle, The application of preconditioning in viscous flows, *J. Comput. Phys.* 105 (1993) 207–223.
- [9] L. Dutto, W. Habashi and M. Fortin, Parallelizable block diagonal preconditioners for the compressible Navier–Stokes equations, *Comput. Methods Appl. Mech. Engrg.* 117 (1994) 15–47.
- [10] G. Hauke and T. Hughes, A unified approach to compressible and incompressible flows, in: F. Navarrina and M. Casteleiro, eds., *Métodos Numéricos en Ingeniería (SEMNI, Barcelona, 1993)*.
- [11] T. Hughes, M. Mallet and A. Mizukami, A new finite element method for CFD: II. Beyond SUPG, *Comput. Methods Appl. Mech. Engrg.* 54 (1986) 341–355.
- [12] T. Hughes and M. Mallet, A new finite element method for CFD: III. The generalized streamline operator for multidimensional advection–diffusion systems, *Comput. Methods Appl. Mech. Engrg.* 58 (1986) 305–328.
- [13] T. Hughes and M. Mallet, A new finite element method for CFD: IV. A discontinuity-capturing operator for multidimensional advective–diffusive systems, *Comput. Methods Appl. Mech. Engrg.* 58 (1986) 329–336.
- [14] A. Jameson, Fast multigrid method for solving incompressible hydrodynamic problems with free surfaces, *AIAA J.* 32(6) (1994) 1175–1182.
- [15] G. Le Beau, S. Ray, S. Aliabadi and T. Tezduyar, SUPG finite element computation of compressible flows with the entropy and conservation variables formulations, *Comput. Methods Appl. Mech. Engrg.* 104 (1993) 397–422.
- [16] G. Le Beau and M. Schultz, Finite element computation of compressible flows with the SUPG formulation, in: M.N. Dhaubhadel, M.S. Engelman and J.N. Reddy, eds., *Advances in Finite Element Analysis in Fluid Dynamics, FED-Vol. 123 (ASME, New York, 1991)* 21–27.
- [17] M. Mallet, A finite element method for CFD, Ph.D. Thesis, Department of Civil Engineering, Stanford University, 1985.
- [18] A. Rizzi and L. Eriksson, Computation of flow around wings based on the Euler equations, *J. Fluid Mech.* 148 (1984) 45–71.
- [19] A. Rizzi and L. Eriksson, Computation of inviscid incompressible flow with rotation, *J. Fluid Mech.* 153 (1985) 275–312.
- [20] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, *SIAM J. Scient. Comput.* 14 (1993) 461–469.
- [21] Y. Saad and M. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Scient. Statist. Comput.* 7 (1986) 856–869.
- [22] F. Shakib, Finite element analysis of the compressible Euler and Navier–Stokes equations, Ph.D. Thesis, Department of Mechanical Engineering, Stanford University, 1988.
- [23] F. Shakib and T. Hughes, A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier–Stokes equations, *Comput. Methods Appl. Engrg.* 89 (1991) 141–219.
- [24] A. Soulaimani and M. Fortin, Finite element solution of compressible viscous flows using conservative variables, *Comput. Methods Appl. Mech. Engrg.* 118 (1994) 319–350.

- [25] M. Storti, N. Nigro and S. Idelsohn, Steady state incompressible flows using explicit schemes with an optimal local preconditioning, *Comput. Methods Appl. Mech. Engrg.* 124 (1995) 231–252.
- [26] M. Storti, N. Nigro and S. Idelsohn, Equal-Order Interpolations: A Unified Approach to Stabilize the Incompressible and Convective Effects, *Comput. Methods Appl. Mech. Engrg.* (1994) submitted.
- [27] M. Storti, C. Baumann and S. Idelsohn, A preconditioning mass matrix to accelerate the convergence to the steady Euler solutions using explicit schemes, *Comput. Methods Appl. Engrg.* 34 (1992) 519–541.
- [28] T. Tezduyar and T. Hughes, Finite element formulations for convection dominated flows with particular emphasis on the compressible Euler equations, *AIAA Paper*, 83-0125, 1983.
- [29] T. Tezduyar, M. Behr, S. Mittal and A. Johnson, Computation of unsteady incompressible flows with the stabilized finite element method—Space–Time formulations, iterative strategies and massively parallel implementations, in: P. Smolinski, W.K. Liu, G. Hulbert and K. Tamma, eds., *New Methods in Transient Analysis*, AMD-Vol. 143 (ASME, New York, 1992) 7–24.
- [30] T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson and S. Mittal, Parallel finite element computation of 3D flows, *IEEE Comput.* (1993) 27–36.
- [31] T. Tezduyar, S. Aliabadi, M. Behr and S. Mittal, Massively parallel finite element simulation of compressible and incompressible flows, *Comput. Methods Appl. Mech. Engrg.* 119 (1994) 157–177.
- [32] T. Tezduyar, J. Liou and D. Ganjoo, Incompressible flow computations based on the vorticity-stream function and velocity–pressure formulations, *Comput. Struct.* 35 (1990) 445–472.
- [33] T. Tezduyar, Stabilized finite element formulations for incompressible flow computations, *Adv. Appl. Mech.* 28 (1991) 1–44.
- [34] B. van Leer, W. Lee and P. Roe, Characteristic time-stepping or local preconditioning of the Euler equations, *AIAA Paper.*, 91-1552-CP, 1991.
- [35] S. Venkateswaran, J. Weiss, C. Merkle and Y. Choi, Propulsion-related flowfields using the preconditioned Navier–Stokes equations, *AIAA Paper.*, 92-3437, 1992.
- [36] R. Vichnevetsky and J.B. Bowles, Fourier analysis of numerical approximations of hyperbolic equations, *SIAM Studies in Applied Mathematics*, 1982.
- [37] O.C. Zienkiewicz, J. Szmelter and J. Peraire, Compressible and incompressible flow: an algorithm for all reasons, *Comput. Methods Appl. Mech. Engrg.* 78 (1990) 105–121.